

# The Proxy Pattern



**Gerald Britton**

Pluralsight Author

@GeraldBritton [www.linkedin.com/in/geraldbritton](http://www.linkedin.com/in/geraldbritton)



# Overview



**Remote proxy**

**Virtual proxy**

**Protection proxy**

**Smart reference proxy**

**DBMSs use them all!**

**Web servers and virtual proxies**



## Demo



### Sensitive information

- Birthdate
- Salary

### AccessControl object

- Employee IDs
- Flag for personal information access

### Client program to access employees

- Use AccessControl object

# Proxy

**Classification: Structural**

**Acts on a real subject**

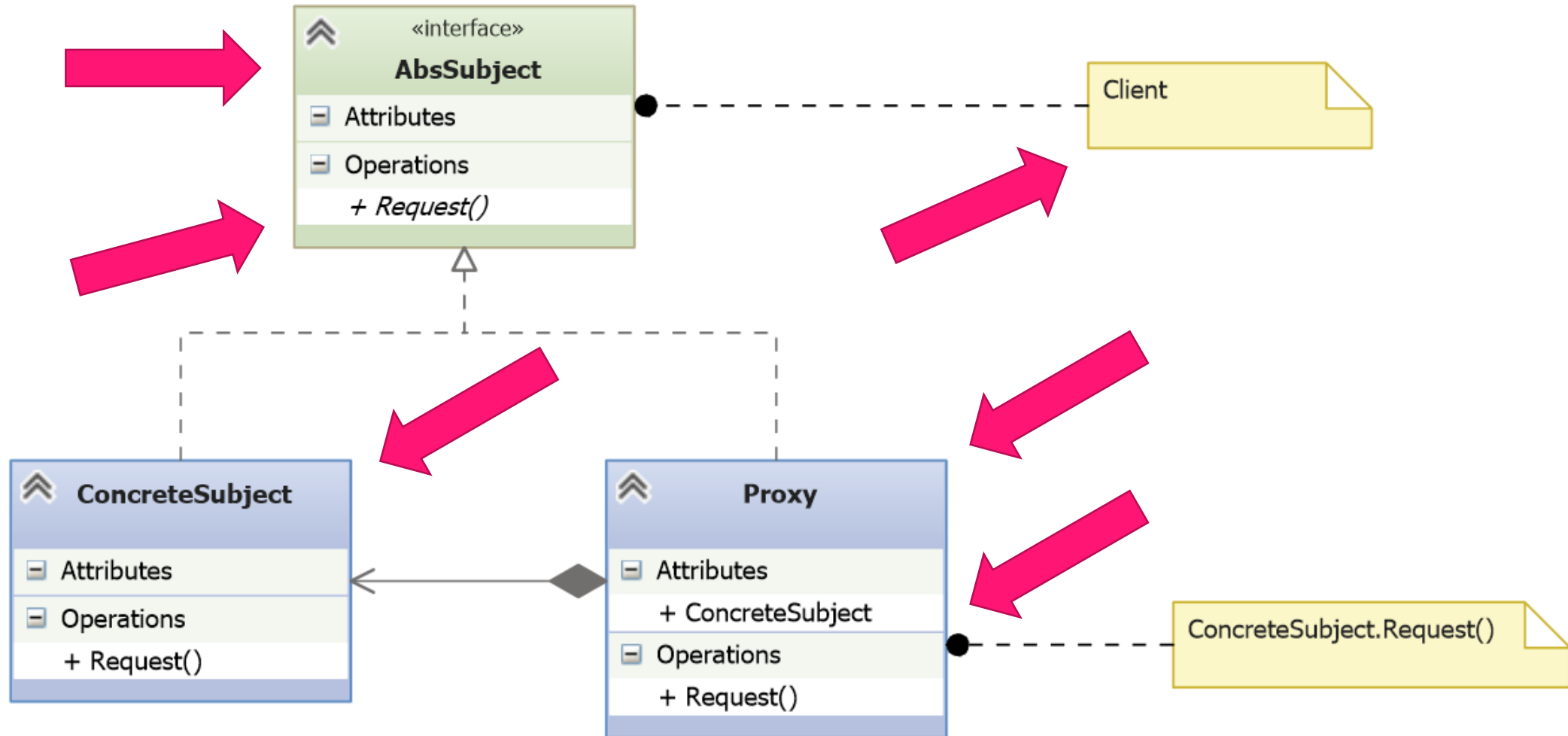
**Keeps a reference to the subject**

**Exposes an identical interface**

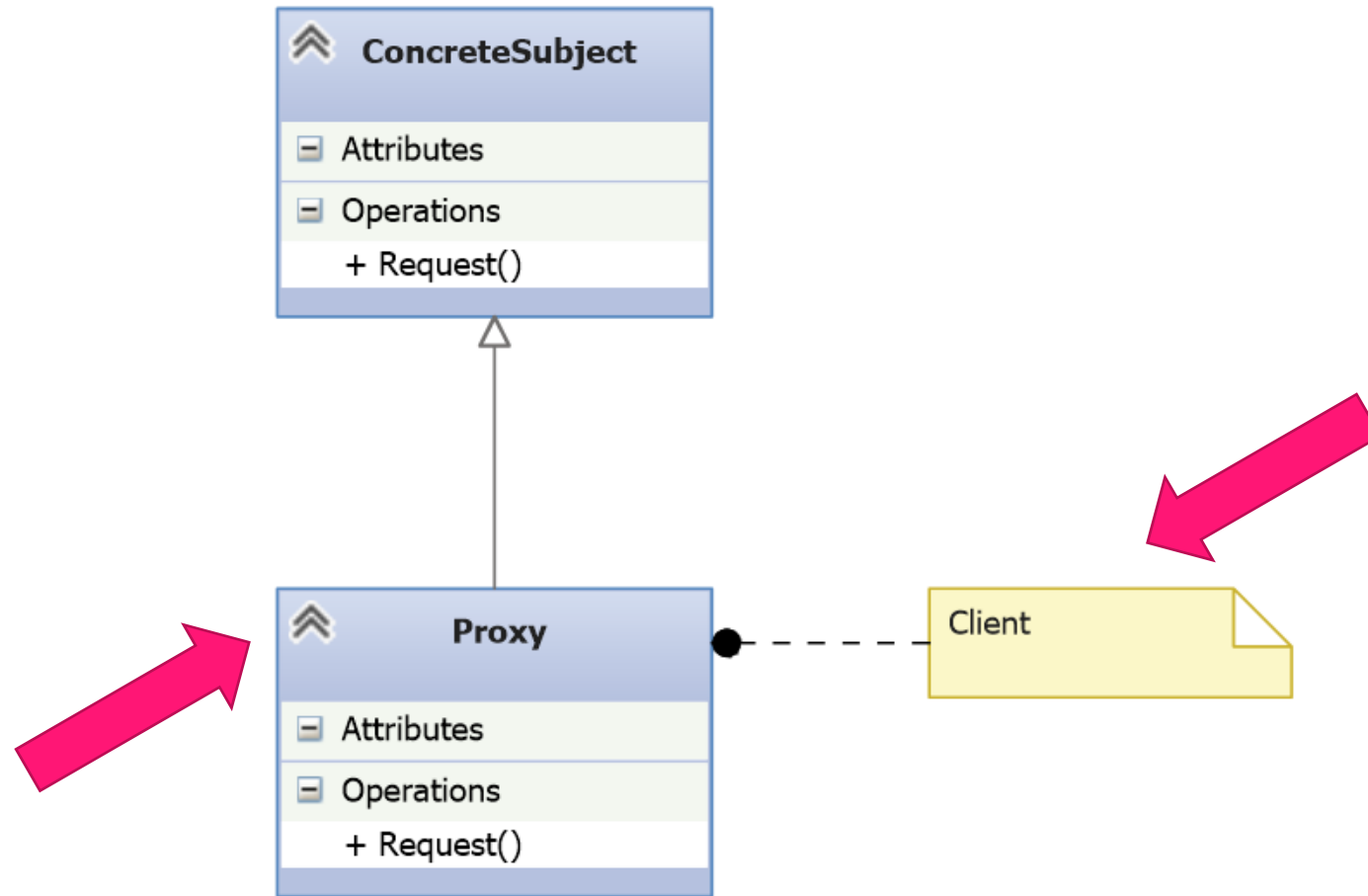
**Controls access to the real subject**



# Proxy Structure



# Simplified Proxy Structure



## Demo



**Implement the protection proxy**

**Create:**

- Subject Abstract Base Class
- Concrete Subject
- Proxy Subject, composed with it

**Use a Factory to get the proxy**

**Test the solution**



# Consequences

**Introduces a level of indirection**

**Protection proxy controls access**

**Virtual proxy and lazy instantiation**

- `@functools.lru_cache`

**Remote proxy hides communication details**

- `pyodbc` for database access

**Smart proxy can add housekeeping**

- Locking

**Open/closed principal**

**Prefer composition over inheritance**





## Summary



**When is the Proxy Pattern applicable?**

**Add controls to an object**

**Obey the open/closed principle**

**Proxies can be used in combination**

